

(57) Abstract: A transaction processing system is provided in which transaction tasks can be modified in real-time without interrupting the operation of the transaction processing system. The system includes intelligent agents [5] that perform selective transaction retrieval from a transaction stream. An agent command and control center [29] enables users to quickly and conveniently construct intelligent agents using simple visual programming techniques. The agents are capable of "eavesdropping" on a transaction stream and retrieving transactions from the stream that fit a given profile. Agents can preferably be modified while they are executing and do not interfere with the underlying transaction processing system. After an agent retrieves a transaction, an Action Manager [26] processes the transaction. Users define the processing that is carried out on transactions preferably using a workflow diagram [22]. The action manager has a visual programming interface that enables users to easily create workflow diagrams by dragging, dropping, and connecting blocks that represent sources of data, analyses, decisions, and actions. The action manager also permits users to redefine workflow while the system is running.

INTELLIGENT TRANSACTION MINING SYSTEM

Field of the Invention

The present invention relates generally to transaction processing systems and, more particularly, to an intelligent transaction mining (ITM) system.

Background of the Invention

Organizations use transaction processing systems to perform a wide variety of functions. These systems are used, e.g., in the banking industry to manage deposits, withdrawals, and transfers of funds among accounts and institutions. The transportation industry uses transaction processing systems to track the arrival and departure of goods at various locations along a delivery route. Packets of data moving across the Internet in world-wide-web applications are also transactions processed by transaction processing systems.

Known transaction processing systems are typically optimized to carry out specific, predefined processing of specific, predefined types of transactions. These processing systems are typically inflexible. Introducing new processing tasks or new transaction types may require a development effort that is significant in terms of cost, time, and risk to both the system and the organizations that rely upon it. Risks include interruption of the system, loss of transactions, and the mishandling of transactions and transactional data.

Transactions may contain a wealth of information that could be used to enhance an organization's effectiveness in a number of ways such as, e.g., improving sales, efficiency, safety, and the quality of products and services. Because transaction processing systems cannot be easily modified, an organization may not be able to make full use of its transactional information in a timely manner.

Data warehousing is a common technique used by organizations to try to derive value from their transactional data. A data warehouse is a repository into which transactional data is dumped. Computer programs can be created to sift through, analyze, and manipulate the warehouse's data. However, relying on data warehouse techniques may be unattractive to an organization for a number of reasons. For instance, creating programs that operate on a data warehouse requires a very high level of skill and significant development time and resources. In addition, warehouses can grow to be enormous, become extremely slow, require large commitments of hardware and human resources, and can easily become unmanageable.

Furthermore, data warehouses do not typically support the capability to respond to transactions promptly in real-time.

Thus a need exists for an improved transaction processing system. Specifically, a need exists for a transaction processing system, in which applications that make use of transactional data can be quickly and easily created. These applications may have a relatively short lifetime (e.g., hours, days, or weeks) and the need for a particular application may arise very suddenly. In addition, a need exists for a transaction processing system that can respond selectively and in real-time to transactions transmitted through high-volume, high-speed channels. Furthermore, a need exists for a transaction processing system that makes use of transactions without jeopardizing transaction data or interrupting the operation of the system.

Brief Summary of the Invention

One object of the present invention is to provide an improved transaction processing system in which applications that make use of transactional data can be quickly and easily created.

Another object of the invention is to provide a transaction processing system that can respond selectively and in real-time to transactions transmitted through high-volume, high-speed channels.

A further object of the invention is to provide a transaction processing system that makes use of transactions without jeopardizing transaction data or interrupting the operation of the system.

These and other objects are accomplished by an inventive transaction processing system (more specifically, an ITM system) in which transaction tasks can be modified in real-time without interrupting the operation of the transaction processing system. The invention utilizes an ITM toolkit, which is a framework and a set of tools for constructing applications that perform intelligent, real-time transaction mining. (The term "transaction mining" refers to the retrieval of transactions of interest from one or more transaction streams for real-time processing.)

In accordance with the invention, intelligent agents perform transaction retrieval. The ITM system includes an Agent Command and Control Center (Agent Center) that enables users to define, test, deploy, monitor, pause, resume, and modify agents. The Agent Center enables users to quickly and conveniently construct intelligent agents using simple visual programming and fill-in-the-blanks techniques. The agents are capable of "eavesdropping" on a transaction stream (e.g., on a WAN, LAN, Intranet, or the Internet) and retrieving transactions from the

stream that fit a given profile. The agents are also capable of autonomous goal-seeking behavior, reasoning, learning, and communicating. They are preferably mobile and can move from one network node to another (i.e., between Agent platforms) to adjust processing loads and the use of network resources. Agents can preferably be modified (or reprogrammed) while they are executing.

The ITM agents do not interfere with the underlying transaction processing systems. Therefore new uses of transactions and transaction information can be easily introduced without interrupting or jeopardizing an organization's underlying core transaction processing systems.

After an ITM agent retrieves a transaction, an ITM Action Manager processes the transaction. Processing may include, e.g., interpreting the transaction, correlating its data with other data (from other transactions, a database, a data warehouse, etc.), and responding to the transaction (e.g., by displaying a message, sending email, generating a web page, updating a database, issuing another transaction, etc.).

Users define the processing that is carried out on transactions by creating a workflow diagram. The Action Manager has a visual programming interface that enables users to create workflow diagrams by, e.g., dragging, dropping, and connecting blocks that represent sources of data (e.g., agents), analyses, decisions, and actions. The Action Manager can also preferably simulate the operation of the entire system, permitting users to test and debug the transaction mining logic and workflow. The Action Manager also permits users to redefine workflow while the system is running.

The blocks in a workflow diagram describe the processing steps that are to be carried out on transactions. For example, a block can represent:

- (1) a simple processing step in its entirety;
- (2) a high level step that can be "exploded" into another workflow diagram (i.e., workflow diagrams are hierarchical); or
- (3) a complex step such as the evaluation of a neural network or a Bayesian probability calculation.

ITM operations are preferably grouped into campaigns. A campaign includes a set of one or more agents and one or more workflow diagrams. The ITM system is capable of running more than one campaign at a time. A Campaign Manager of the ITM system enables operators to start, suspend, resume, and stop the execution of individual campaigns or sets of campaigns.

The ITM system offers numerous advantages over transaction processing systems of the prior art. For instance, the ITM system offers a simple, graphical programming interface that lowers the skill-level required to create, deploy, and operate transaction processing applications. ITM applications can therefore be deployed quickly and are easy to modify and maintain.

In addition, ITM deploys intelligent agents to collect information in real-time. ITM can effect responses to transactions in real-time. ITM operations can be configured to be triggered only by relevant information, thereby avoiding the need to maintain and sift through large volumes of irrelevant data in a data-warehouse.

Moreover, the ITM system's architecture is massively scalable. There is no practical limit to the number of agents or action managers that can be deployed.

Furthermore, the ITM campaign and layout creation and execution preferably occurs within the same environment. There is preferably no "build" step, so changes made to campaigns or layouts are reflected immediately in the way agents retrieve transactions and transaction are processed.

ITM agents are also non-invasive and non-disruptive to core transaction processing systems.

These and other objects and advantages of the present invention will become readily apparent from the following detailed description wherein embodiments of the invention are shown and described by way of illustration of the best mode of the invention. As will be realized, the invention is capable of other and different embodiments and its several details may be capable of modifications in various respects, all without departing from the invention. Accordingly, the drawings and description are to be regarded as illustrative in nature and not in a restrictive or limiting sense with the scope of the application being indicated in the claims.

Brief Description of the Drawings

For a fuller understanding of the nature and objects of the present invention, reference should be made to the following detailed description taken in connection with the accompanying drawings wherein:

FIGURE 1 illustrates a representative ITM system in accordance with the invention;

FIGURE 2 is a screen shot of a sample data sources dialog box in accordance with the invention;

FIGURE 3 is a screen shot of a sample transition source agent configuration dialog box

in accordance with the invention;

FIGURE 4 is a screen shot of a sample gateway agent configuration dialog box in accordance with the invention;

FIGURE 5 is a screen shot of a sample sniffer agent configuration dialog box in accordance with the invention;

FIGURE 6 is a screen shot of a sample rules entry dialog box in accordance with the invention; and

FIGURE 7 is a screen shot of a sample layout diagram in accordance with the invention.

Detailed Description of the Preferred Embodiment

FIGURE 1 illustrates a representative ITM system architecture in accordance with the invention. As shown, one or more Supervisor Platforms 20 run an Agent Command & Control Center module 29, a Campaign Manager module 22 and an Action Manager module 26 (the 'supervisory modules'). Also, one or more Agent Platforms 5 are the deployment location for a plurality of Agents $14_{A1} \dots 14_{Cn}$, a Network Gateway module 12 and an Agent Manager module 16 (the 'agent modules').

Agents of similar type or that work together can be grouped in "villages." (FIGURE 1 shows, e.g., villages A, B, and C.) Agents in a village can be addressed as a group, i.e., a message can be addressed to all agents in a particular group.

Because the supervisory modules and agent modules may simultaneously reside on different platforms, the ITM architecture can be described as a distributed architecture, which is easily scalable. *It should be noted that the supervisory and agent modules can reside on the same computer if so desired.*

As used herein, a platform is any suitable computer processing system with sufficient memory and processing capability to perform the functions described herein. A representative computer platform includes a computer processing unit, a keyboard, a mouse and a display unit. The screen of the display unit is used to present a graphical user interface (GUI) for the user. The GUI is supported by the operating system of the computer and allows the user to use a point and click method of input, e.g., by moving the mouse pointer on the display screen to an icon representing a data object at a particular location on the screen and pressing on the mouse buttons to perform a user command or selection. This type of arrangement also allows the user to "drag and drop" an icon from one position to another on the screen, all in a known manner.

One or more “windows” may be opened up on the computer independently or concurrently as desired.

The Agent Platform 5 is coupled to a transaction bus or stream 10. The transaction bus 10 is shown as a local area network (LAN) in FIGURE 1, which may transmit packets of data using transmission control protocol/Internet protocol (TCP/IP), user datagram protocol (UDP) or any other protocol. It may alternatively comprise other networks such as, e.g., a wide area network (WAN). The bus 10 is coupled to one or more transaction sources. In FIGURE 1, the bus 10 is shown coupled to a host computer 8. The ITM system also preferably contains functionality to simulate transaction sources, in which case a specialized type of agent simulates the host computer 8.

Associated with each agent platform 5 are one or more network gateways 12 that receive transactions from the bus 10 and dispatches them to any number of agents that are linked by intra-process communication to a gateway 12. As used herein, a transaction is a sequence of bit fields, where each field may have different mappings depending upon the type of transaction processing that is being executed in the system. A data dictionary is maintained in the ITM system for mapping the bits of data to transaction fields depending upon the transaction types of the transaction processing system.

One or more agents $14A_1 \dots 14B_1 \dots 14C_n$ may be coupled to the network gateway in each agent platform. In general, an agent includes two operating segments: a sniff segment and a snag segment. The sniff segment includes one or more rules regarding fields of the transactions, such as “extract transaction when field 2 is zero.” The rules for each sniff segment are configured by the user in a manner that is described below with respect to FIGURE 2. It is possible to interface commercially available sniff segments to the system.

Coupled to the sniff segment of each agent is a snag segment, which serves to buffer transactions. If the transaction satisfies the rules set forth in the sniff segment, it is passed up and stored in the snag segment of the agent.

An Agent Manager 16 is coupled to each of the agents on the agent platform. The Agent Manager 16 acts as a proxy for supervisory management directives coming from the Agent Command & Control Center 29 or the Action Manager 26. The Agent Manager 16 controls the propagation and update of rules and the addition and removal of agents $14A_1 \dots 14B_1 \dots 14C_n$ in response to input from a GUI on the Supervisor Platform 20. The Agent Manager 16 also controls the forwarding of transaction data from the various agents in its platforms up to the

Supervisor Platform 20.

Each agent is preferably an object-oriented data structure that includes the following fields: class, methods and properties. An agent that is a subclass of another agent inherits the methods and properties of the parent agent. Every agent has a state, defined by one or more variables, optionally containing a history of previous values of the state variable(s), and one or more parameters. Each agent optionally maintains the current value of the variable(s), each variable's rate of change and the cumulative value over time of each variable. Each agent preferably keeps a record of its age, which is the time elapsed since it was created. Agents can tag a transaction with a trigger time, which is the time when the agent snagged the transaction.

There may be many classes of agents. Each class of agent uniquely identifies the specific behavior of its instance. The three main agent classes in the ITM system are sniffer, gateway and transaction source agent. These agent classes are a subclass of a root agent class called 'ITM Agent'.

The properties and methods of the class ITM Agent are provided in general in the two tables below:

<u>Class:</u>	<u>Properties:</u>
ITM Agent	<ul style="list-style-type: none"> • name of agent • location of agent (address of agent platform)
Sniffer	<ul style="list-style-type: none"> • name of sniffer rule • data dictionary (identify fields of the transaction)
Gateway	<ul style="list-style-type: none"> • address of bus from which agent receives transactions
Transaction Source	<ul style="list-style-type: none"> • address of host computer that posts transactions (used for simulation purposes)

<u>Class:</u>	<u>Methods:</u>	<u>Description:</u>
ITM Agent	Move	Moves the agent to a different platform
	Pause	Pauses the sniff/snag activity of the agent
	Resume	Resumes the sniff/snag activity of the agent
	Start	Starts the agent sniff/snag activity
	Install	Deploys the agent to a platform (a class method)
	Get	Get the value of an agent parameter or variable

	Set	Set the value of an agent parameter or variable
Sniffer	Sniffer rule	Code that determines the transaction field sequence causing the agent to escalate the transaction data
Gateway	Receive transaction	Code to acquire transaction from bus
	Dispatch transaction	Code to write transaction to bus
	Register	Code to register agent with gateway
Transaction Source	Post transaction	Code for posting transaction on the bus

On the Agent Platform, agents are preferably organized into groups of agents that filter different transaction types. Each group preferably contains only the agents that filter a single type of transaction. The Network Gateway copies to each agent only those transactions of the type subscribed to by the agent's group.

Agents are configured in the Agent Command and Control Center 29 using graphical and dialog-based specifications. The Agent Command and Control Center 29 also allows for simulation and debug of agents and deploys or activates the operation of an agent. In addition, the Agent Command and Control Center 29 is capable of monitoring the transaction flow at the agent and controlling the agent using commands such as pause, resume, retire, and re-start.

The Action Manager 26 is a control and analysis module for the ITM system that includes a GUI. In particular, the GUI includes:

- (1) one or more windows for defining an action management workflow diagram or layout;
- (2) a window for defining the network configuration (i.e., the agent platforms and host computer that are tied to the bus 10) and for accessing the agent control centers at each platform; and
- (3) a window for displaying the status of the database network.

The ITM Action Manager 26 permits dynamic modification of the layout so that transaction analysis procedures can be modified in real-time as processing is on-going. ITM Action Manager layouts do not require recompilation to implement changes since all program logic is executed within the G2 interpreted programming environment.

Designing an ITM System

In general, designing an ITM system includes the following four tasks:

- (1) design action management tasks;
- (2) configure simulated transaction source(s);
- (3) configure the network gateway; and
- (4) configure agents.

These tasks can be completed in any order. However, each task (except task (2), which is only required for simulation) must be completed at least once to enable running of the ITM application. The ITM system also allows these steps to be completed during operation. Each task is described in greater detail below.

Design action management tasks

The Action Manager module 26 enables users to develop action management workflow diagrams or “layouts” that define the transaction processing work steps. This component of ITM is a specialization of Gensym’s ReThink™ modeling and simulation package, which is a general purpose graphical programming language. A user can prepare any number of these layouts to represent any sequence of processing steps or tasks, business rules that can act on data from transactions, and any other programmed analysis task. The flow diagram includes a number of boxes or “blocks,” each of which represents a task or activity. Each block can generally be considered as being one of two types:

- (1) Initiate/Terminate (source/sink) blocks (i.e., blocks that are sources of objects that initiate work or activity flow or that are the terminating points for objects passing through the layout); and
- (2) Processing blocks (i.e., blocks that define the work steps, tasks or activities to be performed).

The objects that mediate the tasks defined in an ITM layout can generally be divided into three types:

- (1) User interface objects (i.e., those objects that display user interfaces);
- (2) Business objects (i.e., those objects that perform tasks requesting users interface services from user interface objects and data services from database objects. For example, a transaction is a type of business object); and
- (3) Database objects (i.e., those objects that access databases, the computer file system or some other storage/retrieval entity).

Blocks that are available in the ITM system for use in building action management flow layouts are customized to a particular transaction processing system. However, other blocks, such as merge and copy blocks, are available in a base package and are provided as a generic set of blocks for the ITM system. Blocks may include tasks capable of reasoning in real-time with regard to events that occur in the system.

ITM is designed to simplify the integration of other Gensym software packages by inserting this software into ITM blocks. The Gensym software that can be inserted into the blocks includes Gensym's BayesOn-Line, NeurOn-Line, NeurOn-Line Studio, CDG, GDA, Operations Expert or any Gensym software assets.

Configure Simulated Transaction Source Agents:

As mentioned above, the bus 10 in the ITM system may be coupled to any device that initiates transactions. Due to the portable design of the ITM architecture, which is implemented in software, the transaction processing system may be easily simulated to verify its operation and optimize its performance prior to deploying the application on-line. A host computer (such as host computer 8) may be coupled to the bus 10 to provide a representative series of transactions at a rate identical to that to be received in the actual operation of the ITM transaction processing system. Alternatively, a transaction source agent can be used to simulate the host machine 8. The transaction source agent can be configured in a "data sources" dialog box 100, a sample of which is shown in FIGURE 2. The data sources dialog box is displayed by selecting 'View-Data Sources' from the ITM main menu.

The functions available for management of transaction source agents include those shown in the table below. These functions are accessed from the 'button bar' 102 in the dialog box 100.

<u>Function</u>	<u>Description</u>
Show Map	Show the host network map
Launch	Launch the currently selected agent
Start	Start the currently selected agent
Replace Agent	Replace the currently selected agent (if it is deployed)
Retire Agent	Retire the currently selected agent (if it is deployed)
Reset Agent	Reset the currently selected agent (if it is deployed)
Create New Agent	Create a new agent (create an entry in the spreadsheet)
Configure Agent	Configure the agent
Delete Agent	Delete the agent from the spreadsheet
Go to Agent (in G2)	Go to the agent's representation in G2

Refresh View	Refresh the agent view
--------------	------------------------

The transition source agent configure window 120 (an example of which is shown in FIGURE 3) is accessed by selecting a "Configure Agent" button in the dialog box 100 of FIGURE 2. In the window 120, various agent characteristics such as those shown can be specified.

Configure network gateway agents:

In this step, the gateway is configured to receive data from the attached network. This configuration step connects the ITM application to the network by entering data necessary to identify the type of network protocol and the type of transaction to be sniffed. Any networking protocol supported in the International Organization for Standardization/Open Systems Interconnection (ISO/OSI) seven-layer model can be accommodated by the addition of different Network Gateway classes that implement methods capable of sniffing any type of data packet. A typical network gateway agent configuration dialog 140 is shown in FIGURE 4.

Configure sniffer agents:

Agents are configured in the Agent Command and Control Center 29 using the dialog 160 shown in FIGURE 5 or one from the ITM layout diagram 190, a sample of which is shown in FIGURE 7. The FIGURE 5 dialog is used to access the dialog to enter the sniffer rule or rules and other sniffer agent parameters such as:

- (1) the type of transaction to sniff; and
- (2) the agent deployment platform (of which there may be many as indicated in FIGURE 5).

The dialog accessible from the ITM layout may include the dialog of FIGURE 5 or modifications of this dialog to display only those parameters of interest to users developing the action management layout. The functions available in the sniffer agent dialog can include those shown in the table below.

<u>Function</u>	<u>Description</u>
Save	Save the agent information
Add Transaction	Add a new transaction (as a class) to be sniffed
Specify Agent Platform	Specify the platform onto which the agent is to be deployed
Show Rules	Show the rule entry dialogue

Add Rules	Add a new rule
Show Map	Show the map of agent platforms

Rules are entered in a rule entry dialog 180, a sample of which is shown in FIGURE 6. Each rule can act on any data that is available to the agent, including the agent's own variables (variables that the agent maintains) or parameters and data that is available in the transaction. Users interactively create rules with a series of clicks. After the rule is entered, the user selects a "Generate Rule" option button to create the agent-sniffing rule.

The agents of the present invention are developed as extensions to the Agent Development Environment (ADE) used for agent management.

Operating the ITM System

In general, operating the ITM system includes the following steps:

- (1) deploy transition source agents (if simulating), sniffer agents, agent managers and gateway agents; and
- (2) run system
 - a) using simulated transactions (if in simulation mode)
 - b) using actual transactions.

In the first step, the transition source agents, sniffer agents, agent managers and gateway agents are deployed. Once the configuration information for each of these components has been provided, they may be deployed. Deployment involves (1) generating the agent code (including the agent rules that were entered by the user), (2) transferring compiled code to agent platform(s), and (3) executing this code. When an agent is deployed, the move and start methods are called on the agent.

In the second step, the ITM system is run. As previously discussed, it may be desirable to forward simulated transactions over the bus 10 for purposes of debug and analysis of the operation of the ITM system. Simulation may be useful before the product is released to a customer, or alternatively by a customer who wishes to customize his transaction processing system. Simulation is not a required step of the invention. Simulation is conducted by activating a transition source agent, which emits transactions for analysis by the other components of the ITM application. During simulation, these other components of the ITM application act just as they would if actual transactions were used.

In either the simulation mode or an actual transaction mode, the transition source agent

(when in the simulation mode) or host computer (when in the actual use mode) forwards transactions over the bus. The network gateway agent copies transactions to the sniffer segments of the agents and, if the transaction passes the sniffer agent's rule, the transactions are escalated up to the action management flow diagram. The action management flow diagram specifies the real-time dynamic analysis that will be performed on transactions that are escalated. Objects and links within the GUI are highlighted as they are accessed to show the progress of this analysis.

The inventive ITM system can be used in a variety of different fields including, e.g., the banking and transportation industries. The invention provides dynamic, interactive, real-time control in substantially any type of transaction processing system. With the ITM system, an agent can be constructed to perform virtually any type of task conveniently by visually specifying the agent behavior, which is translated into the appropriate code.

Having described embodiments of the present invention, it should be apparent that modifications can be made without departing from the scope of the present invention.

Claims

1. A transaction processing system, comprising:
means for processing transactions based on at least one transaction processing task; and
means for modifying the at least one transaction processing task in real-time without interrupting transaction processing.
2. The transaction processing system of Claim 1 further comprising at least one agent for selectively retrieving transactions from a transaction stream, wherein behavior of said at least one agent is characterized by a method, and wherein the method can be dynamically modified during transaction processing to modify the at least one transaction processing task.
3. The transaction processing system of Claim 2 further comprising means for dynamically adding additional agents to the transaction processing system during operation.
4. The transaction processing system of Claim 2 wherein the at least one transaction processing task and the at least one agent are configured using object-oriented visual programming techniques.
5. The transaction processing system of Claim 1 further comprising a graphical user interface, wherein specification or modification of the at least one transaction processing task is performed using the graphical user interface.
6. The transaction processing system of Claim 1 wherein the at least one transaction processing task is capable of reasoning about events in real-time.
7. The transaction processing system of Claim 1 wherein said system is scalable.
8. The transaction processing system of Claim 1 further comprising a transition source agent for simulating a transaction data source.
9. A transaction processing system comprising:
means for selectively retrieving transactions from a transaction stream, said means including an agent characterized by parameters and methods, and wherein the parameters or methods of the agent are dynamically modifiable during operation of the transaction processing system without interrupting the operation; and
means for processing retrieved transactions.

10. The transaction processing system of Claim 9 wherein the means for selectively retrieving and the means for processing are implemented in distributed computer architecture.

11. The transaction processing system of Claim 9 wherein said means for processing retrieved transactions include an action manager, and wherein said action manager and said agent reside on different computers.

12. The transaction processing system of Claim 9 wherein the means for selectively retrieving include another agent, and wherein said agent and said another agent reside on separate computers.

13. The transaction processing system of Claim 9 further comprising a transition source agent for simulating a transaction data source.

14. A transaction processing system comprising:
network gateway coupled to a transaction stream;
at least one dynamically modifiable agent for selectively retrieving transactions from said network gateway; and
an action manager for processing transactions retrieved by said at least one agent.

15. The transaction processing system of Claim 14 further comprising an agent manager for updating and coordinating operation of the at least one agent.

16. The transaction processing system of Claim 14 further comprising an agent command and control center for supervising functions of said at least one agent.

17. The transaction processing system of Claim 14 wherein said agent command and control center can dynamically add new agents to the transaction processing system during operation.

18. The transaction processing system of Claim 14 wherein said action manager and said at least one agent reside on different computers.

19. The transaction processing system of Claim 14 further comprising a second agent for selectively retrieving transactions from said network gateway, and wherein said agent and said second agent reside on separate computers.

20. The transaction processing system of Claim 14 wherein the action manager and the agent are configured using object-oriented visual programming techniques.
21. The transaction processing system of Claim 14 wherein said action manager includes at least one transaction processing task that is modifiable in real-time without interrupting transaction processing.
22. The transaction processing system of Claim 14 wherein the at least one agent is characterized by parameters and methods, and wherein the parameters or methods are dynamically modifiable during operation of the transaction processing system.
23. The transaction processing system of Claim 14 further comprising a transition source agent for simulating a transaction data source.
24. A method of intelligent transaction mining, comprising:
selectively retrieving transactions from a transaction stream;
processing retrieved transactions based on at least one transaction processing task;
and
modifying the at least one transaction processing task in real-time without interrupting transaction processing.
25. The method of Claim 24 wherein agents are used for selectively retrieving transactions and wherein the method further comprises dynamically adding additional agents during transaction processing.
26. The method of Claim 24 further comprising using object-oriented visual programming techniques to configure agents used for selectively retrieving transactions from a transaction stream and to configure tasks used in processing retrieved transactions.
27. The method of Claim 24 further comprising using a graphical user interface to specify or modify a task used in processing retrieved transactions.
28. The method of Claim 24 further comprising a simulating a transaction data source.
29. The method of Claim 24 wherein modifying the at least one transaction

processing task in real-time comprises modifying a workflow diagram describing transaction processing.

30. The method of Claim 24 wherein an agent is used for selectively retrieving transactions and wherein the agent is characterized by a parameter and a method, and wherein the parameter or method of the agent is dynamically modifiable.

31. An agent used in a transaction processing system for selectively retrieving transactions to be processed, comprising:

a sniff segment including at least one rule for determining which transactions are to be retrieved; and

a snag segment for receiving and storing transactions satisfying the at least one rule.

32. The agent of Claim 31 wherein said agent is an object-oriented data structure.

33. A method of configuring an action manager for processing transactions in a transaction processing system, comprising preparing a workflow diagram representing processing steps to be performed on the transactions using a graphical user interface, said workflow diagram including a plurality of blocks representing the processing steps and data sources.

ITM Architecture

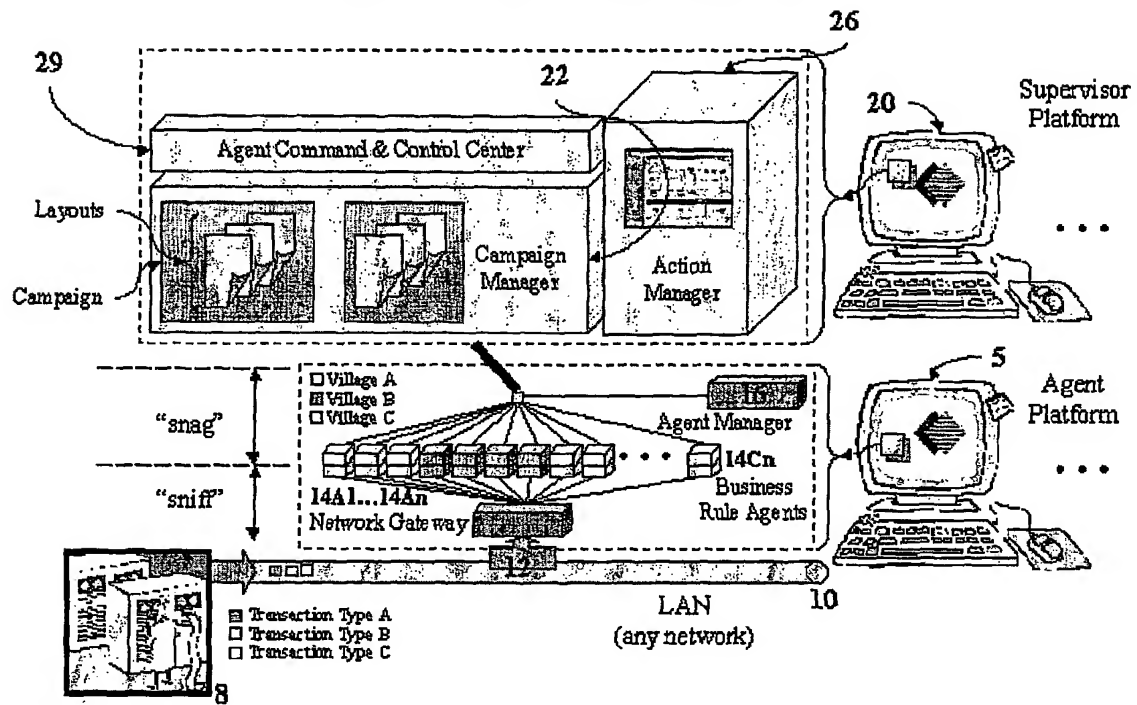


Figure 1

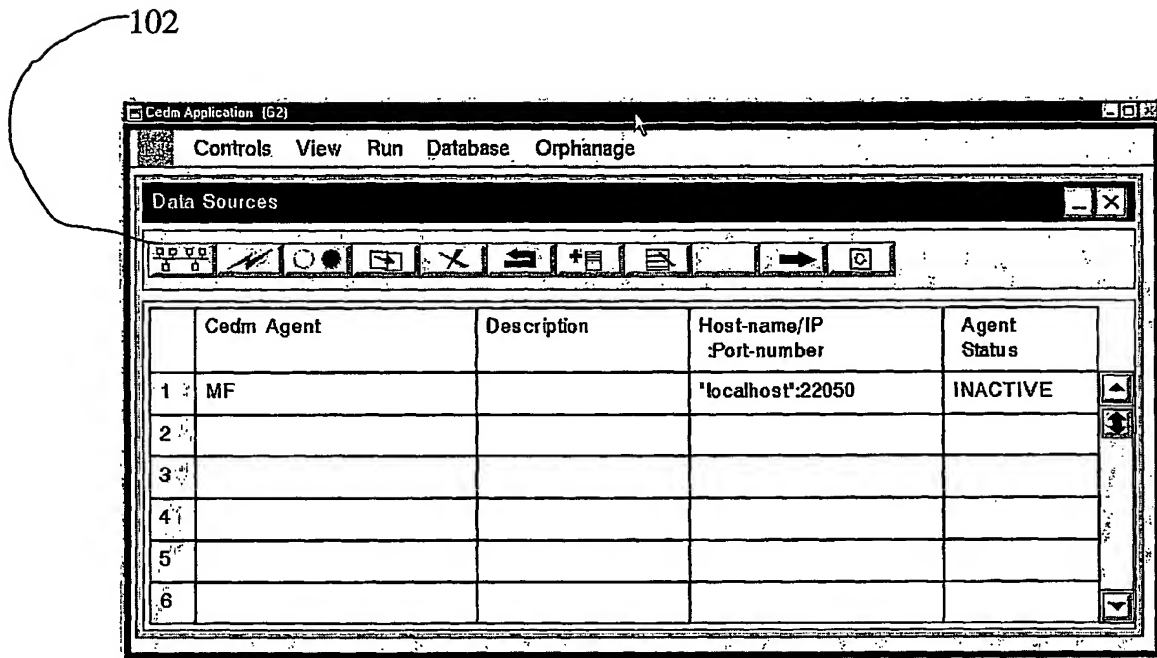


Figure 2

100

Cedm Application (62)

Controls View Run Database Orphanage

Configuration Dialog for Main-Frame-Agent

Name	MF
Display Java Agent	<input checked="" type="checkbox"/>
Record #	0
Fixed Format	YES
Machine/Host ID	localhost
Port Number	22050
Transaction Class	DBM-BANK-TRANS
Multicast Hostname	172.16.255.255
Multicast Port	1234
File Name	C:\cedm\data\bank.dat
Time Delay	500
Description	

Figure 3

Cedm Application (G2)

Controls View Run Database Orphanage

Configuration Dialog for Gateway-Agent

Configuration Dialog for Gateway-Agent

Name	GATEWAY
Display Java Agent	YES
Record #	0
Fixed Format	YES
Machine/Host ID	localhost
Port Number	22050
Multicast Hostname	172.16.255.255
Multicast Port	1234
Port to listen to...	1235
Description	

Figure 4

140

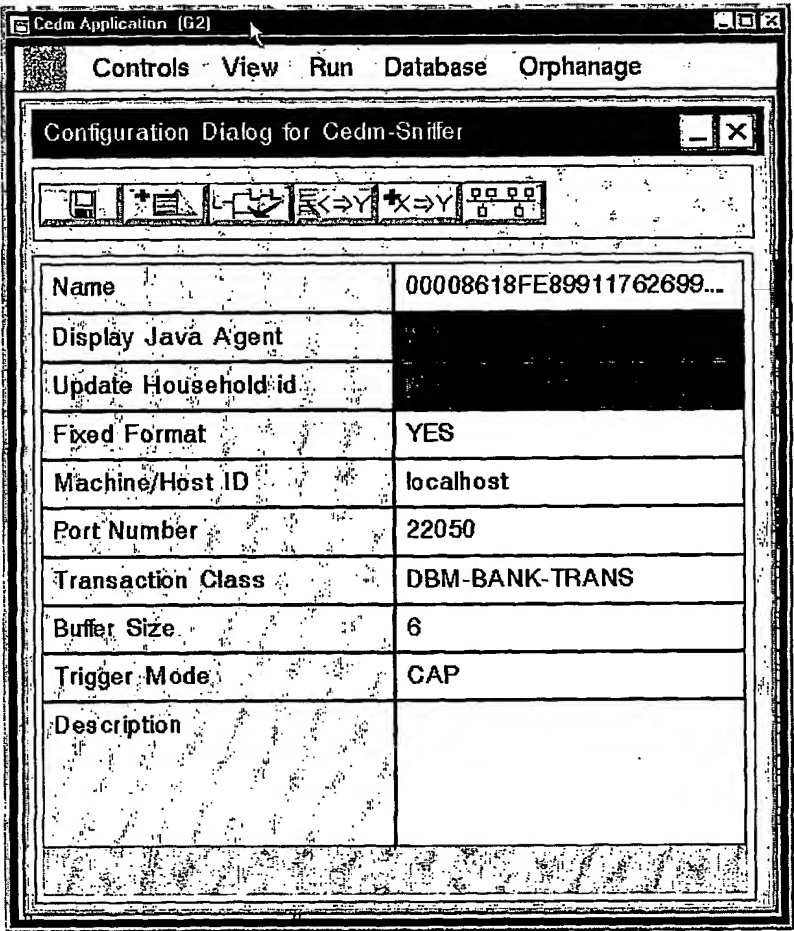


Figure 5

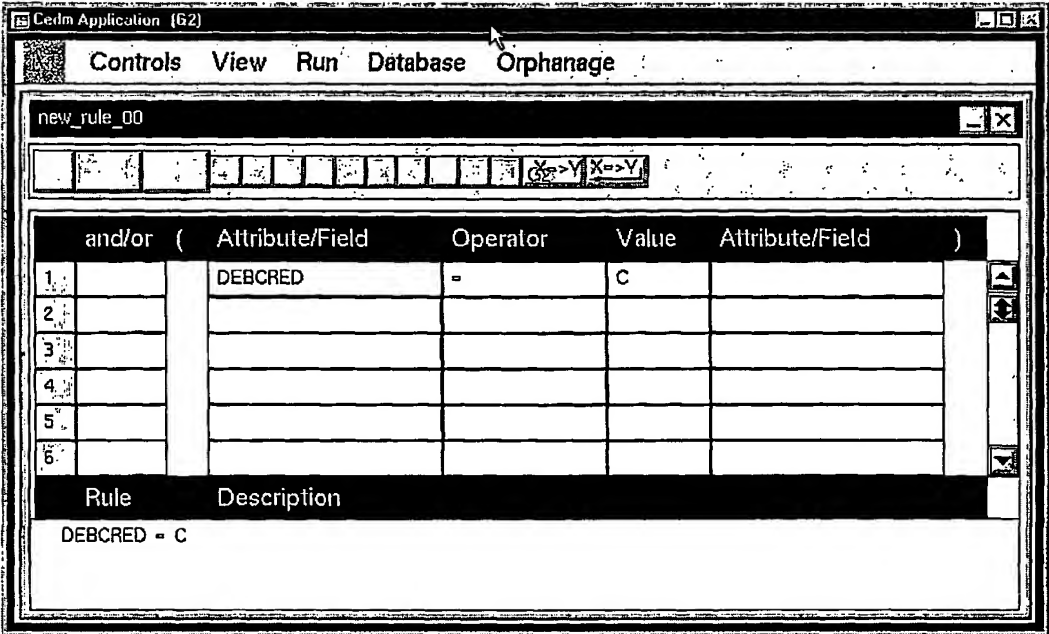


Figure 6

180

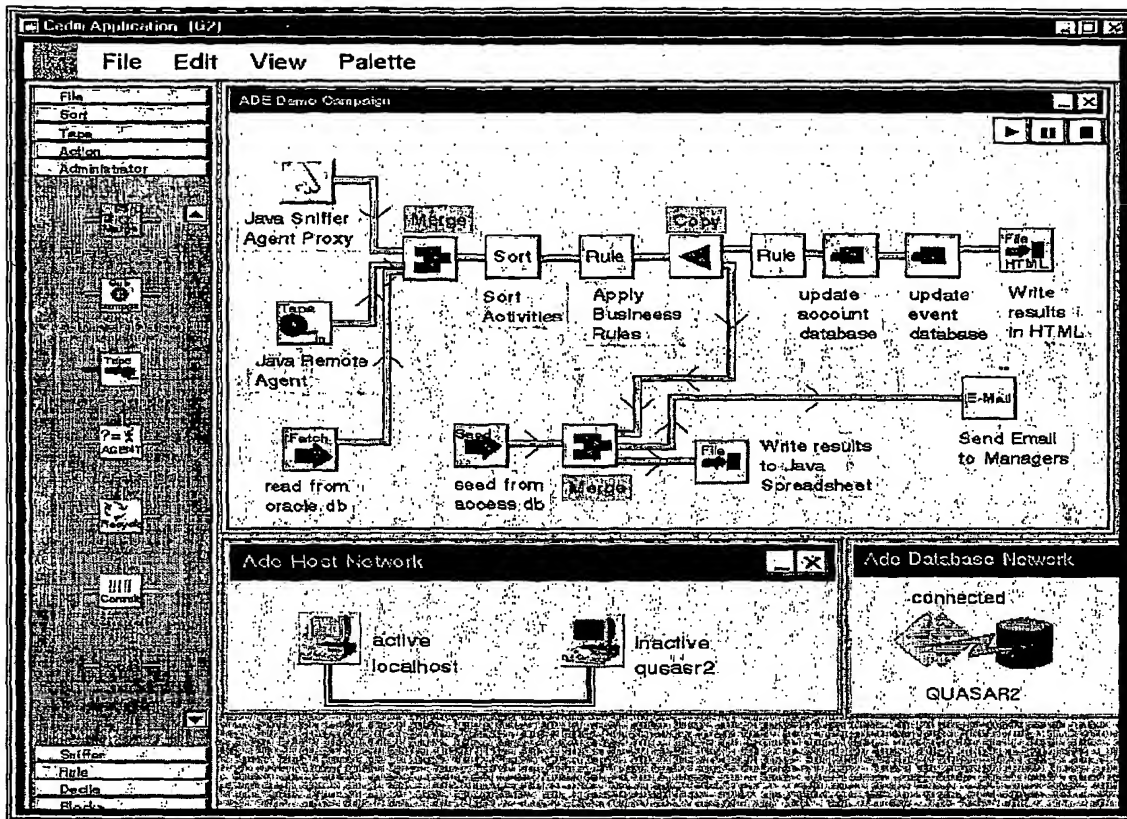


Figure 7

190

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US01/11711

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 17/30

US CL :707/1

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/1, 103, 10, 202, 201, 102, 104; 709/203

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

West, CAS online

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y, P	US 6,138,143 A (GIGLIOTTI et al.) 24 October 2000, See fig 3.	1-33
Y, P	US 6,122,642 A (MEHOVIC) 19 September 2000, See abstract.	1-33
Y, P	US 6,157,927 A (SCHAEFER et al.) 05 December 2000, See abstract.	1-33

☐ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

19 MAY 2001

Date of mailing of the international search report

14 JUN 2001

 Name and mailing address of the ISA/US
 Commissioner of Patents and Trademarks
 Box PCT
 Washington, D.C. 20231.

Facsimile No. (703) 305-3230

Authorized officer

SANJIV SHAH

James R. Matthews

Telephone No.

(703) 305-8355